Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel

Nicolas Delfosse[1,2,3] and Gilles Zémor[4]
[1]*IQIM, California Institute of Technology, Pasadena, California 91125, USA*
[2]*Department of Physics and Astronomy, University of California, Riverside, California 92507, USA*
[3]*Microsoft Quantum and Microsoft Research, Redmond, Washington 98052, USA*
[4]*Mathematical Institute, IMB, UMR 5251, Bordeaux University, France*

(Received 6 November 2019; accepted 6 April 2020; published 9 July 2020)

Surface codes are among the best candidates to ensure the fault tolerance of a quantum computer. In order to avoid the accumulation of errors during a computation, it is crucial to have at our disposal a fast decoding algorithm to quickly identify and correct errors as soon as they occur. We propose a linear-time maximum likelihood decoder for surface codes over the quantum erasure channel. This decoding algorithm for dealing with qubit loss is optimal both in terms of performance and speed.

DOI: 10.1103/PhysRevResearch.2.033042

I. INTRODUCTION

Surface codes [1–3] are one of the leading candidates to ensure the fault tolerance of a quantum computer. Error correction is based on the measurement of local operators on a lattice of qubits. The measurement outcome, called the syndrome, is then processed by the decoding algorithm which uses this information to infer the error which occurred. In order to avoid the accumulation of errors during computation, it is essential for the decoder to be fast.

The quantum erasure channel [4,5] is the noise model that represents loss or leakage outside the computational space in multilevel systems. Erasures are ubiquitous in various quantum hardware due to photon loss in optical elements or long-distance quantum communication [6–8], atom loss [9], or leakage in trapped ions [10,11].

The quantum erasure channel also provides a theoretical framework in high-energy physics to explore the ADS-CFT correspondence between anti-de Sitter (ADS) spaces and conformal field theories (CFT) which allows one to reconstruct an erased region on the boundary of the model [12]. Finally, the performance of quantum error correcting codes against erasures relates to the ability to perform fault-tolerant gates on encoded qubits [13,14].

The loss of a qubit is equivalent to applying a random Pauli error to this qubit, while giving, as additional data, the position of the error. For stabilizer codes, this extra information reduces the decoding problem to solving a linear system, which can be done with cubic complexity. In the particular case of surface codes, the syndrome of an error is a set of vertices of a lattice and decoding amounts to finding a set of paths connecting these vertices by pairs or to the boundary. One could pick two vertices and connect them by a path and repeat until all the syndrome vertices are matched. This would lead to a quadratic complexity. This strategy was adopted by Dennis *et al.* to decode Pauli errors [2] or by Barrett and Stace in the case of a combination of Pauli errors and erasures [15].

In the present work, we propose a linear-time maximum likelihood decoder for erasures over surface codes. This is optimal both in terms of performance and in terms of complexity. Our algorithm can be used with any surface code, with arbitrary genus, and any type of boundary [16–18], including hyperbolic codes [19–21]. In the case of Pauli errors the efficient maximum likelihood decoder obtained by Bravyi *et al.* [22], only applies to a restricted set of surfaces.

In the rest of the paper, we describe our decoding algorithm and we prove that it is a maximum likelihood decoder and that its complexity is linear. We first consider Kitaev's codes. Then we generalize the approach to surfaces with boundaries, more relevant for practical purposes [3].

II. KITAEV'S CODES

Kitaev's codes [1] are obtained by imposing local constraints on qubits placed on a closed surface. Since only the combinatorial structure of the surface matters, we denote by (V, E, F) such a surface with vertex set V, edge set E, and face set F. These three sets are assumed to be finite. An edge $e \in E$ is a pair of distinct vertices $e = \{u, v\}$. A face is a region of the surface homeomorphic to a disk and delimited by a set of edges. We represent a face by the set of edges lying on its boundary. We assume that the graph (V, E) has neither loops nor multiple edges. We also suppose that its dual is well defined and satisfies the same properties.

Consider the Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes E}$. Each qubit is indexed by an edge e, and the Pauli operator acting on this qubit as the matrix X, Y, or Z and acting trivially elsewhere is denoted respectively by X_e, Y_e, or Z_e. *Kitaev's code* is defined to be the ground space of the Hamiltonian

$$-\sum_{v\in V} X(v) - \sum_{f\in F} Z(f),$$

where $X(v) = \prod_{v \in e} X_e$ and $Z(f) = \prod_{e \in f} Z_e$. The operators $X(v)$ and $Z(f)$ generate a group S, the *stabilizer group*, which fixes the code space. Elements of S are called *stabilizers*. The Z stabilizers, which are products of face operators $Z(f)$, are the operators of $\{I, Z\}^{\otimes E}$ whose support is a trivial cycle of G. By *cycle*, we mean here a subset of edges of G which meets every vertex an even number of times. A cycle is said to be *trivial* if it lies on the boundary of a set of faces. In the same way, X stabilizers correspond to trivial cycles of the dual graph. The correction of Pauli errors is based on the measurement of the generators $X(v)$ and $Z(f)$ which tells us whether or not the error commutes with these operators. The outcome of this measurement is called the *syndrome* of the error. Errors with a trivial syndrome, meaning that they commute with all the stabilizers, can be seen as operators acting on the code space and are called *logical operators*. For instance, stabilizers are trivial logical operators. Nontrivial logical operators correspond to nontrivial cycles in the graph G or its dual.

III. MAXIMUM LIKELIHOOD DECODER FOR QUBIT LOSS

The *quantum erasure channel* is one of the most simple noise models. Each qubit is lost, or erased, independently with probability p. Such a loss can be detected and the missing qubit is then replaced by a totally mixed state $\mathcal{I}/2$. Writing $\mathcal{I}/2 = \frac{1}{4}(\rho + X\rho X + Y\rho Y + Z\rho Z)$, we see that this new qubit can be interpreted as the original state which suffers from a Pauli error I, X, Y, or Z chosen uniformly at random. The set of lost qubits is denoted by $\mathcal{E}$. The encoded state is subjected to a random uniform Pauli error P whose support is included in $\mathcal{E}$. Denote this condition by $P \subset \mathcal{E}$.

Just like when dealing with Pauli noise, one can then measure the stabilizer generators $X(v)$ and $Z(f)$ and try to recover the error P from its syndrome. The main difference with Pauli channels is the additional knowledge of the erasure pattern $\mathcal{E}$. Since operators of S act trivially on the code space, the goal of the decoder is to identify the coset PS of the error, knowing the set $\mathcal{E}$ and the syndrome σ of P. The optimal strategy, called *maximum likelihood decoding* (MLD), is to maximize the conditional probability $\mathbb{P}(PS|\mathcal{E}, \sigma)$.

To illustrate how the knowledge of the erasure $\mathcal{E}$ simplifies the decoding problem, assume that we found an error $\tilde{P} \subset \mathcal{E}$ whose syndrome matches σ. Both errors P and $\tilde{P}$ have the same syndrome, hence $\tilde{P}$ and P differ in a logical operator $L \subset \mathcal{E}$, trivial or not. Due to the fact that errors $Q \subset \mathcal{E}$ are uniformly distributed, $\mathbb{P}(QS|\mathcal{E}, \sigma)$ is proportional to the number $|(QS) \cap \mathcal{E}|$ of Pauli errors of that coset that are included in $\mathcal{E}$. This number depends only on the number $|S \cap \mathcal{E}|$ of stabilizers having support inside $\mathcal{E}$, which shows that all the cosets are equiprobable. Therefore, MLD consists simply of returning an error coset $\tilde{P}S$ such that $\tilde{P} \subset \mathcal{E}$ and the syndrome of $\tilde{P}$ is equal to a given σ. This proves the following.

Lemma 1. Given an erasure $\mathcal{E} \subset E$ for a surface code and a measured syndrome σ, any coset $\tilde{P}S$ of a Pauli error $\tilde{P} \subset \mathcal{E}$ of syndrome σ is a most likely coset.

The same argument applies for any stabilizer code.

Algorithm 1. MLD for surface codes.

```
Require: A surface $G = (V, E, F)$, an erasure $\mathcal{E} \subset E$ and the
  syndrome $\sigma \subset V$ of a $Z$ error.
Ensure: A $Z$ error $P$ such that $P \subset \mathcal{E}$ and $\sigma(P) = \sigma$.
 1: Construct a spanning forest $F_{\mathcal{E}}$ of $\mathcal{E}$.
 2: Initialize $A$ by $A = \emptyset$.
 3: While $F_{\mathcal{E}} \neq \emptyset$, pick a leaf edge $e = \{u, v\}$ with pendant
    vertex $u$, remove $e$ from $F_{\mathcal{E}}$ and apply the two rules:
 4:     (R1) If $u \in \sigma$, add $e$ to $A$, remove $u$ from $\sigma$ and flip $v$ in $\sigma$.
 5:     (R2) If $u \notin \sigma$ do nothing.
 6: Return $P = \prod_{e \in A} Z_e$.
```

IV. LINEAR-TIME MAXIMUM LIKELIHOOD DECODER

We now propose a fast algorithm that returns such a most likely coset for Kitaev's codes. We detail the construction of the Z part of the error with Algorithm 1, illustrated n Fig. 1. The same algorithm will be applied to the dual graph to recover the X part of the error.

Only measurements of operators $X(v)$ can detect a Z error. The syndrome of a Z error P is thus the subset $\sigma(P) \subset V$ of vertices v such that $X(v)$ anticommutes with this error. Equivalently, it is the set of vertices surrounded by an odd number of qubits supporting an error Z. In order to translate our decoding problem into a graphical language, denote by $\partial(A)$ the set of vertices that a subset $A \subset E$ encounters an odd number of times and call it the *boundary* of A. The syndrome of the Z-error pattern supported on A is exactly $\partial(A)$. Given $\mathcal{E} \subset E$ and $\sigma \subset V$, we are looking for a subset of edges $A \subset \mathcal{E}$ such that $\partial(A) = \sigma$.

Paradoxically, an obstacle to a linear-time complexity is the presence of cycles in $\mathcal{E}$. Although cycles increase the number of paths from a vertex to another and potentially make it easier to find one, they also make it easier to make suboptimal choices. Our basic idea is not to try to sequentially find paths that pair the syndrome vertices together but instead to shrink recursively the set of edges on which we have yet to make a decision. To this end we select a *spanning forest* $F_{\mathcal{E}}$ inside $\mathcal{E}$, that is, a maximal subset of edges of $\mathcal{E}$ that contains no cycle and spans all the vertices of $\mathcal{E}$. If $\mathcal{E}$ is a connected graph,

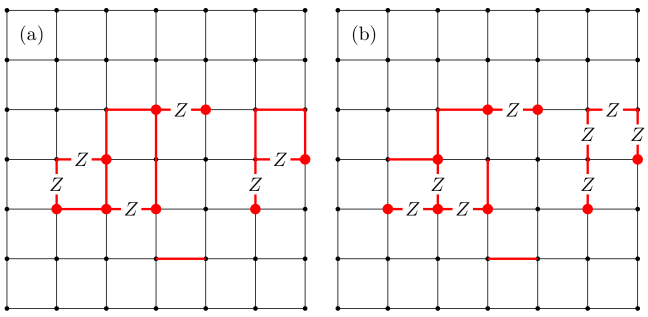

FIG. 1. (a) A square lattice of the torus. Red thick edges mark the set $\mathcal{E}$ of erased qubits which support some Z error. Its syndrome is indicated by large red nodes. (b) A spanning forest $F_{\mathcal{E}}$ [thick red lines in (b)] is constructed. Then, starting from the leaves, an error included in the $F_{\mathcal{E}}$ is constructed using the syndrome. This provides a correct estimation of the error up to a stabilizer.

then $F_\mathcal{E}$ is also connected and is called a *spanning tree*. Such a forest can be found in linear time [23].

Equipped with the forest $F_\mathcal{E}$ that contains all the syndrome vertices, we can now find the required subset A very efficiently. Starting with the empty set, we construct A, by applying recursively the following rules.

(R1) Pick a *leaf*, that is, an edge $e = \{u, v\}$ connected to the forest through only one of its two endpoints, say v. The vertex u is called a *pendant vertex*. Assume first that $u \in \sigma$, then we add the edge e to the set A and we *flip* the vertex v. By flipping, we mean that v is added to the set σ if $v \notin \sigma$ and it is removed from σ in the case $v \in \sigma$. Then, e is removed from the forest $F_\mathcal{E}$.

(R2) In the case when $u \notin \sigma$, this edge is simply removed from $F_\mathcal{E}$ and A is kept unchanged.

Through these two steps, we peel the forest $F_\mathcal{E}$ until only an empty set remains. The construction of the set A is then complete. This procedure relies on the following obvious remark, stated as a lemma to emphasize the role of the two rules applied in Algorithm 1.

Lemma 2 (leaf alternative). Let A be a subset of edges of a tree T. If $e = \{u, v\}$ is a leaf with pendant vertex u, then (R1) either $u \in \partial(A)$ and $e \in A$, or (R2) $u \notin \partial(A)$ and $e \notin A$.

This strategy is guaranteed to end after a finite number of steps. It remains to show that it returns a set A such that $A \subset \mathcal{E}$ and $\partial(A) = \sigma$. We must verify that such a set A exists and that the peeling process does not depend on the order in which leaves of the forest are removed. This is done in the proof of Theorem 1.

Theorem 1. For surface codes with bounded degree and faces of bounded size, applying Algorithm 1 to the graph and to its dual produces a linear-time maximum likelihood decoder.

During step 3 of the algorithm, a naive approach would be to look for a leaf by running over the forest at each round but this strategy would lead to a superlinear complexity. However, we can ensure linear complexity by running over the whole forest and precomputing a list of leaves. For a bounded degree graph, this list can then be updated in constant time at each round when an edge is removed from the forest.

Proof. Finding a spanning forest has a linear cost, then our algorithm runs over each edge of the forest only once, leading to a linear-time complexity overall. We have to prove that the set A, constructed by this algorithm, satisfies the claimed properties. The fact that $A \subset \mathcal{E}$ is immediate. Only the condition $\partial(A) = \sigma$ deserves some attention. First, we will show that, for any choice of $F_\mathcal{E}$, there exists a set $A \subset F_\mathcal{E}$ such that $\partial(A) = \sigma$ and that this set is unique. Then we will see that applying (R1) and (R2), starting from the leaves, indeed constructs this set A.

Existence: There exists a subset B such that $B \subset \mathcal{E}$ and $\partial(B) = \sigma$ since σ is the syndrome of an error. We will reroute the paths contained in B to squeeze this subset inside $F_\mathcal{E}$ without changing its boundary. Let $x_1, \dots, x_m$ be the edges of $B \backslash F_\mathcal{E}$. By maximality of the forest $F_\mathcal{E}$, adding any extra edge x_i to $F_\mathcal{E}$ creates a cycle $\gamma_i \subset F_\mathcal{E} \cup \{x_i\}$. In order to remove x_1 from the set B, replace B by $B_1 = B \Delta \gamma_1$ where Δ denotes the symmetric difference of these two sets of edges. Then, $x_1 \notin B_1$, only edges of $F_\mathcal{E}$ are added to B and $x_2, \dots, x_m$ are untouched. By repeating this transformation, one creates a sequence $B_{i+1} = B_i \Delta \gamma_i$ such that $B_{i+1} \subset T_\mathcal{E} \cup \{x_i, \dots, x_m\}$ for $i = 1, \dots, m$. The last set, B_m, is included in $F_\mathcal{E}$. Taking the symmetric difference with a cycle γ_i preserves the boundary, i.e., $\partial(B_i) = \partial(B)$ for all i. This proves that the set B_m satisfies both conditions $B_m \subset F_\mathcal{E}$ and $\partial(B_m) = \sigma$. This is our set A.

Uniqueness: If there exist two such subsets A and A', their symmetric difference $A \Delta A'$ is a subset of the forest which has a trivial boundary $\partial(A \Delta A') = \emptyset$ meaning that $A \Delta A'$ is a cycle. Since this cycle is in a forest, it can only be the empty set, proving that $A = A'$.

Now that existence and uniqueness of A are established, we see that the alternative offered by Lemma 2 can only end with the set A. The result of our algorithm is independent of the order in which we pick the leaves in step 3 by uniqueness of A. The existence of A guarantees that our algorithm finds this set after peeling the whole forest. ■

V. SURFACES WITH BOUNDARIES

Kitaev's surface codes were generalized to surfaces with boundaries [16,17], leading to a key simplification for the experimental realization of topological codes. Adapting Algorithm 1 to these codes presents two difficulties. First, the syndrome depends on the type of boundary and second, the spanning forest has to be grown in a way that depends on the boundary type. In what follows, we present a generalization of Algorithm 1 to the case of surface with boundaries.

We use the formalism of [18] that encompasses both generalizations of Kitaev's codes [16,17]. We consider a *surface $G = (V, E, F)$ with boundary*, which means that some edges belong to a unique face. On the boundary, some edges and their endpoints are declared to be *open*. We denoted by $\breve{E}$ (respectively, $\breve{V}$) these open sets and by $\mathring{V} = V \backslash \breve{V}$ and $\mathring{E} = E \backslash \breve{E}$ the nonopen sets. Qubits are placed on nonopen edges and the *generalized surface code* is defined as the ground space of the Hamiltonian

$$-\sum_{v \in \mathring{V}} X(v) - \sum_{f \in F} Z(f),$$

where $X(v) = \prod_{v \in e, e \in \mathring{E}} X_e$ and $Z(f) = \prod_{e \in f, e \in \mathring{E}} Z_e$. No qubit is placed on an open edge and open vertices do not support any operator $X(v)$.

Consider an erasure $\mathcal{E} \subset \mathring{E}$ which comes with a Pauli error affecting erased qubits. Again, we focus on the correction of Z-type errors. Open vertices do not support any measurement $X(v)$. Hence, the syndrome of a Z error of support $A \subset \mathring{E}$ is given by the restriction of $\partial(A)$ to nonopen vertices. Denote by $\mathring{\partial}(A) \subset \mathring{V}$ this restricted boundary. The missing information on open vertices makes it impossible to reconstruct the error starting from those vertices. We must find a way to peel the whole forest using only nonopen vertices. In order to be sure that the peeling algorithm is not stuck before removing all the edges of the forest, we will grow the forest starting from open vertices and peel it the other way around as depicted in Fig. 2.

In Algorithm 2, we adapt the construction of the spanning forest. First, let us explain a simple strategy to find a spanning tree of a connected graph $H = (V, E)$. For a

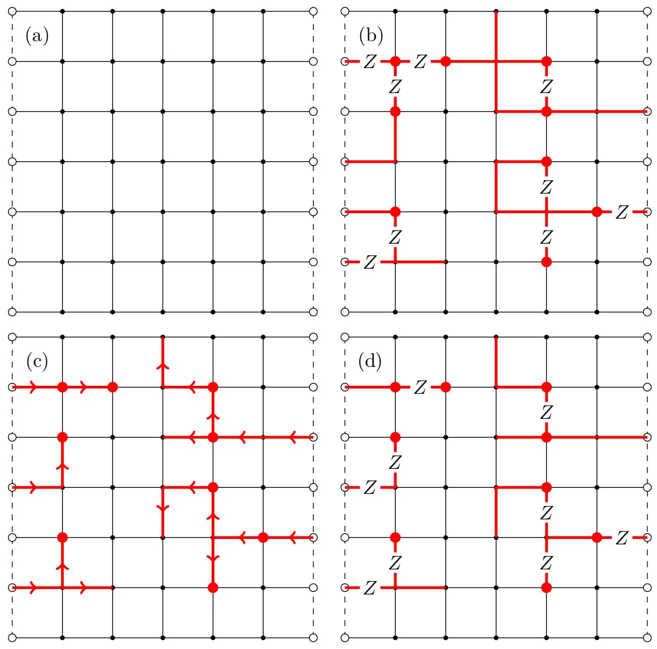

FIG. 2. (a) Surface code with boundaries. White nodes and dashed lines represent open vertices and open edges. (b) Red thick lines indicate an erasure $\mathcal{E}$ with a Z error and its syndrome given by the large red vertices. (c) A spanning forest $F_\mathcal{E}$, with open vertices as a seed. Arrows show the way the forest is grown. (d) The error is estimated by reversing the arrows.

general graph, applying this method to all the connected components produces a spanning forest. Our starting point is a tree T that contains only a single arbitrary vertex v of H and no edge. We grow T by adding edges incident to the tree that connect T with a vertex of H that does not already belong to T. After adding $|V|-1$ edges, one gets our spanning tree.

In Algorithm 2, we will grow a spanning forest of a graph $H=(V,E)$ equipped with a marked subset of vertices $O\subset V$ that we call the *seed*. The spanning tree of a connected component containing a vertex $v_O\in O$ is constructed starting with this vertex v_O. Then, just as before we add edges that reach new vertices but we also require that these newly reached vertices do not belong to O. Based on this strategy, the spanning tree obtained is connected to the boundary in at most one open vertex. If the connected component does not contain any seed vertex, the previous method applies.

Algorithm 2. MLD for surfaces with boundaries.

```
Require: A surface $G=(V,E,F)$ with boundaries, an erasure
    $\mathcal{E}\subset\mathring{E}$, and the syndrome $\sigma\subset\mathring{V}$ of a $Z$ error.
Ensure: A $Z$ error $P$ such that $P\subset\mathcal{E}$ and $\sigma(P)=\sigma$.
 1: Construct a spanning forest $F_\mathcal{E}$ of $\mathcal{E}$ with seed $\check{V}\cap V(\mathcal{E})$
      where $V(\mathcal{E})$ is the set of vertices incident to $\mathcal{E}$.
 2: Initialize $A$ by $A=\emptyset$.
 3: While $F_\mathcal{E}\neq\emptyset$, pick a leaf edge $e=\{u,v\}$ with pendant
      vertex $u\in\mathring{V}$, remove $e$ from $F_\mathcal{E}$, and apply the two rules:
 4:    (R1) If $u\in\sigma$, add $e$ to $A$, remove $u$ from $\sigma$, and flip $v$ in $\sigma$.
 5:    (R2) If $u\notin\sigma$ do nothing.
 6: Return $P=\prod_{e\in A}Z_e$.
```

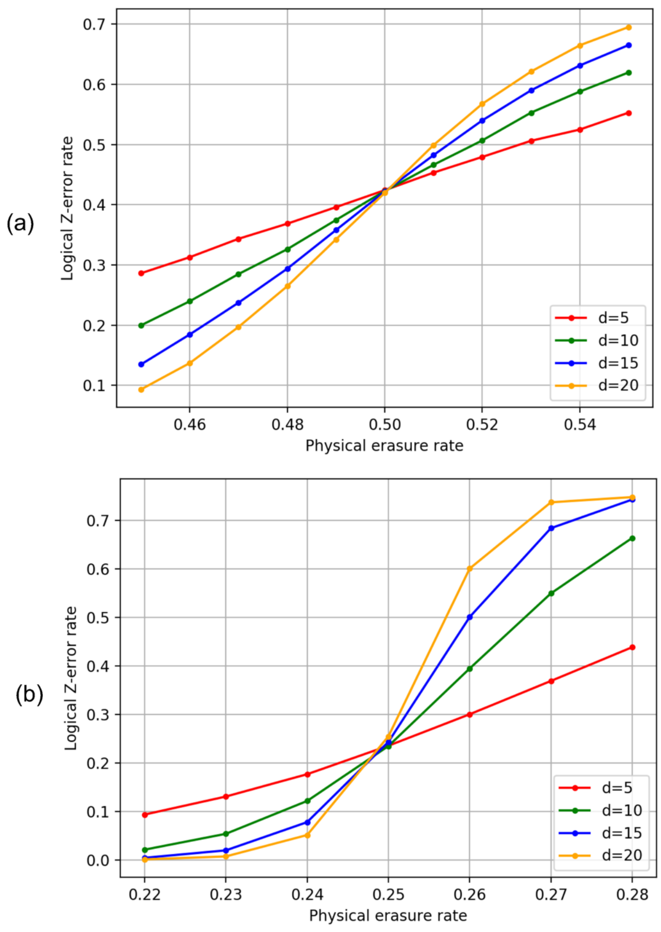

FIG. 3. Monte Carlo simulation of the peeling decoder for (a) the 2D toric code and (b) the 3D toric code. Each data point is obtained from 10^5 decoding trials.

Theorem 2. For generalized surface codes with bounded degree and faces of bounded size, applying Algorithm 2 to the graph and to its dual produces a linear-time maximum likelihood decoder.

Proof. Existence and uniqueness of the set A follow from the same argument as in the proof of Theorem 1 after replacing cycles by relative cycles. Recall that a relative cycle is a subset of edges that meets each nonopen vertex an even number of times. The space of relative cycles of graph is studied for instance in Sec. 4.1 of [18].

Then, Lemma 2, which provides the recursive construction of the error, is used in an identical way. We only need to make sure that the pendant vertices u picked in step 3 are not open. Our algorithm picks these vertices by reversing the construction of the forest with open vertices as a seed. This guarantees that one can peel the whole forest. ■

Figure 3 shows the results of our numerical simulation of the Peeling decoder for the two-dimensional (2D) toric code and the three-dimensional (3D) toric code with erasure on edges. For these two codes, we observed a threshold of 50% and 24.9%, which matches the optimal threshold predicted by bond percolation theory [24].

VI. CONCLUSION

Despite the presence of inconvenient short cycles, we managed to design an optimal decoding algorithm for correction of erasures for arbitrary surface codes. In the case of classical error correction, studying erasure decoding paved the way for better and better decoders in the case of more complicated channels. We may hope that in the quantum setting, solving the decoding problem for the erasure channel may similarly lead towards improved decoders for more complicated noise models and other families of codes.

A serious obstacle to decoding quantum Low Density Parity Check (LDPC) codes is also the presence of short cycles in their Tanner graph. How to deal with them in general remains widely open [25–28]. It is crucial to consider such generalizations that may allow for fault-tolerant universal quantum computation with a considerably reduced overhead [29–32].

One could also consider the correction of losses assuming imperfect gates and measurements [33,34] or in the context of linear optical quantum computing where photon losses are a major obstacle [6,35–37].

ACKNOWLEDGMENTS

N.D. thanks Jonas Anderson for his comments on a preliminary version of this work. N.D. acknowledges funding provided by the Institute for Quantum Information and Matter, an NSF Physics Frontiers Center (NSF Grant No. PHY-1125565) with the support of the Gordon and Betty Moore Foundation (GBMF-2644).

[1] A. Y. Kitaev, Ann. Phys. **303**, 27 (2003).
[2] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).
[3] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A **86**, 032324 (2012).
[4] M. Grassl, T. Beth, and T. Pellizzari, Phys. Rev. A **56**, 33 (1997).
[5] C. H. Bennett, D. P. DiVincenzo, and J. A. Smolin, Phys. Rev. Lett. **78**, 3217 (1997).
[6] E. Knill, R. Laflamme, and G. J. Milburn, Nature (London) **409**, 46 (2001).
[7] B. Bell, D. Herrera-Martí, M. Tame, D. Markham, W. Wadsworth, and J. Rarity, Nat. Commun. **5**, 5480 (2014).
[8] F. Ewert, M. Bergmann, and P. van Loock, Phys. Rev. Lett. **117**, 210501 (2016).
[9] J. Vala, K. B. Whaley, and D. S. Weiss, Phys. Rev. A **72**, 052318 (2005).
[10] R. Fazio, G. M. Palma, and J. Siewert, Phys. Rev. Lett. **83**, 5385 (1999).
[11] L.-A. Wu, M. S. Byrd, and D. A. Lidar, Phys. Rev. Lett. **89**, 127901 (2002).
[12] F. Pastawski, B. Yoshida, D. Harlow, and J. Preskill, J. High Energy Phys. 06 (2015) 149.
[13] S. Bravyi, D. Poulin, and B. Terhal, Phys. Rev. Lett. **104**, 050503 (2010).
[14] F. Pastawski and B. Yoshida, Phys. Rev. A **91**, 012305 (2015).
[15] S. D. Barrett and T. M. Stace, Phys. Rev. Lett. **105**, 200502 (2010).
[16] M. H. Freedman and D. A. Meyer, Found. Comput. Math. **1**, 325 (2001).
[17] S. B. Bravyi and A. Y. Kitaev, arXiv:quant-ph/9811052.
[18] N. Delfosse, P. Iyer, and D. Poulin, arXiv:1606.07116.
[19] M. H. Freedman, D. A. Meyer, and F. Luo, *Mathematics of quantum computation* (Chapman and Hall, London, 2002), pp. 287–320.
[20] G. Zémor, in *Proceedings of the Second International Workshop on Coding and Cryptology (IWCC 2009)* (Springer-Verlag, Berlin, 2009), pp. 259–273.
[21] N. P. Breuckmann and B. M. Terhal, IEEE Trans. Inf. Theory **62**, 3731 (2016).
[22] S. Bravyi, M. Suchara, and A. Vargo, Phys. Rev. A **90**, 032326 (2014).
[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms* (MIT, Cambridge, MA, 2009).
[24] T. M. Stace, S. D. Barrett, and A. C. Doherty, Phys. Rev. Lett. **102**, 200501 (2009).
[25] D. J. C. MacKay, G. Mitchison, and P. L. McFadden, IEEE Trans. Inf. Theory **50**, 2315 (2004).
[26] D. Poulin and Y. Chung, Quantum Inf. Comput. **8**, 987 (2008).
[27] N. Delfosse and G. Zémor, Quantum Information & Comput. **13**, 793 (2013).
[28] J.-P. Tillich and G. Zémor, IEEE Trans. Inf. Theory **60**, 1193 (2013).
[29] D. Gottesman, Quantum Inf. Comput. **14**, 1338 (2014).
[30] O. Fawzi, A. Grospellier, and A. Leverrier, in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, New York, 2018), pp. 743–754.
[31] E. Campbell, Quant. Sci. Technol. (2019).
[32] N. P. Breuckmann and V. Londe, arXiv:2001.03568.
[33] A. C. Whiteside and A. G. Fowler, Phys. Rev. A **90**, 052316 (2014).
[34] M. Suchara, A. W. Cross, and J. M. Gambetta, in *2015 IEEE International Symposium on Information Theory (ISIT)* (IEEE, New York, 2015), pp. 1119–1123.
[35] M. A. Nielsen, Phys. Rev. Lett. **93**, 040503 (2004).
[36] D. E. Browne and T. Rudolph, Phys. Rev. Lett. **95**, 010501 (2005).
[37] K. Kieling, T. Rudolph, and J. Eisert, Phys. Rev. Lett. **99**, 130501 (2007).